이동 컴퓨팅 환경에서 요구 패턴 분석을 기반으로 하는 캐쉬 대체 전략

(A Cache Replacement Strategy based on the Analysis of Request Patterns in Mobile Computing Environments)

이 윤 장 † 신 동 천 ††
(Yoon-Jang Lee) (Dong-Cheon Shin)

요 약 낮은 대역폭을 갖는 이동 컴퓨팅 환경에서 캐싱은 요구 경쟁을 감소시킴으로서 응답시간을 향상시킬수 있는 유용한 방법이다. 전통적인 캐쉬 기반의 시스템에서는 히트율을 향상시키는 것이 일반적으로 빠른 응답시간을 위한 주요한 관심사중의 하나였다. 그렇지만, 이동 컴퓨팅 환경에서는 히트율 뿐만 아니라 미스 비용의 고려도 필요하다. 본 논문에서는 풀 기반의 데이타 분산 시스템에서 새로운 캐쉬 대체 전략을 제시하고 시뮬레이션을 통하여 성능을 평가한다. 제시한 전략은 인기도와 대기 시간을 함께 고려하여 인기도와 대기 시간을 곱한 값 중에서 가장 작은 값을 갖는 페이지를 대체 페이지로 선정한다.

키워드 : 이동 컴퓨팅, 데이타 방송, 캐쉬 대체 전략

Abstract Caching is a useful technique to improve the response time by reducing contention of requests in mobile computing environments with a narrow bandwidth. In the traditional cache based systems, to improve the hit ratio has been usually one of main concerns for the fast response time. However, in mobile computing environments, it is necessary to consider the cost of cache miss as well as the hit ratio. In this paper, we propose a new cache replacement strategy in pull-based data dissemination systems. Then, we evaluate performance of the proposed strategy by a simulation approach. The proposed strategy considers both the popularity and the wating time together, so the page with the smallest value of multiplying popularity by waiting time is selected as a victim.

Key words : mobile computing, data broadcast, cache replacement strategy

1. 서 론

최근들어 통신 기술의 급속한 발전은 컴퓨팅 환경뿐만 아니라 통신 환경까지도 급속도로 변화시켰다. 이러한 통신 환경의 변화는 데이타 분산 기반 시스템(data dissemination-based system)과 같이 제한된 대역폭(bandwidth)에서 다수의 클라이언트 요구를 수용할 수 있도록 하기 위한 시스템에 대한 연구를 촉진시키고 있으며, 이러한 분산 기반 시스템 환경은 클라이언트 수나 데이타베이스 크기의 대규모화, 통신환경의 비대칭성(asymmetry), 요구하는 데이타의 높은 집중도등으로 특징지울 수 있다[1,2].

분산 기반 시스템이 취할 수 있는 전송 기법은 크게 푸쉬(push)와 풀(pull) 기반으로 나눌 수 있다[1,3]. 푸쉬 기반의 데이타 전송 방식에서 클라이언트는 데이타에 대한 어떠한 요구도 서버로 전달하지 않지만 풀 기반의 데이타 전송 방식에서는 클라이언트가 필요로 하는 데이타를 직접 서버에게 요청하고 서버는 특정 기준에 의해 페이지를 선별하여 방송하게 된다. [4]에서는 이러한 두 가지 기법의 장점을 취하기 위한 방법으로 두 가지를 혼용하는 하이브리드(hybrid) 기반 기법을 소개하고 있다.

유선 환경보다 불특정 다수의 클라이언트를 가진 이동 컴퓨팅 환경에서는 한정된 대역으로 인해 초기 유선 환경과 같이 풀 기반 응용보다는 푸쉬 기반 응용에 관한 연구가 진행되어 왔다[3,5]. 그러나, 유선 환경에서도

† 비 회 원 : 중앙대학교 정보시스템학과
lyoon74@hotmail.com
†† 종신회원 : 중앙대학교 산업과학대학 정보시스템학과 교수
dcshin@cau.ac.kr
논문접수 : 2002년 2월 20일
심사완료 : 2003년 5월 7일

네트워크 인프라와 통신기술의 발전으로 NOD, VOD, 대화형 TV와 같은 풀 기반 응용에 관한 연구가 새롭게 진행되고 있다[6,7]. 이러한 추세는 무선 통신 기술의 발전으로 무선 환경에서도 풀 기반의 응용에 대한 연구를 촉발시키고 있다.

한편, 대규모 풀 기반 환경에서 캐쉬는 클라이언트 간의 페이지에 대한 요구 경쟁을 감소시키고, 페이지의 응답 시간을 개선할 목적으로 사용된다. 이러한 캐쉬의 효과는 적용할 환경에 적합한 캐쉬 대체 전략을 사용하였을 때 극대화 될 수 있다. 지금까지 제안된 캐쉬 대체 전략은 대체 페이지 선정 기준으로 어떠한 요소를 고려하고 있는가에 따라서 푸쉬 기반 혹은 풀 기반 환경에 적합한지가 결정된다.

데이타 분산 기반 시스템은 다수의 클라이언트에게 데이타를 보내주기 때문에, 각 페이지의 응답 시간은 틀려질 수 있으며, 특히, 이동 컴퓨팅 환경과 같이 전송 대역폭의 크기가 제한된 환경에서는 이러한 결과가 더욱 두드러지게 나타난다. 이러한 이유로 인해서 히트율이 높아도 응답 시간이 큰 페이지의 미스로 인해 평균 응답 시간이 길어지는 결과를 초래하게 된다. 따라서, 이동 컴퓨팅 환경에서의 데이타 분산 기반 시스템은 다수의 클라이언트의 요구 경쟁과 제한된 대역을 특징으로 하기 때문에 히트율과 미스 비용을 함께 고려하는 대체 전략이 필요하다.

표 1은 기존에 제안된 캐쉬 대체 전략이 어떠한 요소를 고려하여 대체 페이지를 선정하는지를 요약한 것이다. 푸쉬 환경에서 제안된 PIX는 각 페이지의 접근 확률과 방송 주기내에 방송되는 페이지의 방송 빈도를 이용하여 대체할 페이지를 선택하는 알고리즘이다[5]. 각 페이지의 PIX 값은 해당 페이지의 접근 확률(P) ÷ 주기 내 방송 빈도(X)로 구하고, PIX 값이 작은 페이지를 캐쉬에서 대체할 페이지로 선정한다. 따라서, 방송 빈도가 상대적으로 매우 적거나 페이지의 접근 확률이 상대적으로 매우 큰 페이지가 캐쉬에 남게 되어 히트율과 대기 시간을 함께 고려하는 방법이라 할 수 있다. PT는 PIX의 단점을 보완하기 위해 푸쉬 환경에서 제안되었다. PT는 페이지 접근 확률 × (다음 방송 시간 − 현재 시간)을 이용하여 계산하고, PT 값이 가장 작은 페이지가 대체할 페이지로 선정된다[11].

Gray 전략[8]은 CF[9]와 LRU[10]의 단점을 보완하고 히트율과 대기 시간 모두를 고려하기 위해 CF와 LRU를 조합하여 대체할 페이지를 선택하는 알고리즘이다. LRU를 위해서 Gray 알고리즘에서는 {black, gray, white} 세 가지 상태를 접근할 수 있는 모든 페이지에

표 1 캐쉬 대체 전략의 비교

	히트율	미스 비용
FIFO(first in first out)[10]	×	×
LFU(least frequently used)[10]	○	×
LRU(least recently used)[10]	○	×
LRU K[11]	○	×
LRFU(least recently/frequently used)[13]	○	×
PIX[5]	○	○
PT[5]	○	○
CF(closest first)[9]	×	○
Gray[8]	○	○

부여한다. black은 현재 단계에서 요청된 페이지를 의미하고, gray는 이전 단계에서 요청된 페이지를, white는 요청되지 않은 페이지를 의미한다. 현재 단계에서는 페이지가 요청되는 경우 black 표시를 하고, 페이지 미스가 발생할 경우 white로 표시된 페이지 중에서 CF를 이용하여 대체할 페이지를 선택한다.

미스 비용을 고려한 기존의 캐쉬 대체 전략은 이동 컴퓨팅 환경에서 푸쉬 기반의 분산 기반 시스템에 사용하기 위해 제안되었다. PIX, PT, Gray, CF 등은 방송 디스크[5]를 가정하였기 때문에, 서버 방송 스케줄링에 종속적이다. 그러나, 풀 기반 환경에서는 이러한 방송 디스크를 적용하기 어렵다. 즉, 데이타를 주기적이며 반복적으로 보내지 않고 요구에 따라 비주기적인 방송 형태를 취하는 풀 환경에서 기존에 제안된 미스 비용을 고려한 캐쉬 대체 전략은 사용될 수 없다.

본 논문에서는 이동 컴퓨팅 환경에서 풀 기반의 데이타 분산 기반 시스템에 적합한 새로운 캐쉬 대체 전략을 제안한다. 기존 캐쉬 대체 전략의 한계를 극복하기 위하여 제안한 캐쉬 대체 전략은 클라이언트의 요구 패턴 분석을 기초로 대체할 페이지를 선정하도록 하였다. 따라서, 제안된 캐쉬 대체 전략은 각 클라이언트의 요구 패턴을 보다 정확히 반영함으로써, 클라이언트 요구 패턴 변화가 심한 환경에도 무리없이 적용할 수 있다. 한편, 시뮬레이션을 통하여 히트율, 평균 응답시간, 최악 응답시간을 척도로 제안한 대체 전략의 성능평가를 시도하였다.

본 논문의 구성은 다음과 같다. 2장에서는 인기도 및 대기 시간과 응답 시간과의 관계를 도출하여 요구 패턴 분석의 토대를 기술한다. 3장에서는 요구 패턴 분석의 토대를 바탕으로 새로운 캐쉬 대체 전략을 제안한다. 4장에서는 제안된 캐쉬 대체 전략의 성능 평가를 하고 5장에서 결론을 맺는다.

2. 요구 패턴 분석

2.1 인기도

인기도란 사용자의 관심에 따라 특정 페이지가 접근되는 빈도를 수치화 한 값이다. 인기도는 크게 단기(short-term) 인기도와 장기(long-term) 인기도로 나눌 수 있다. 단기 인기도는 시간의 흐름에 따른 클라이언트의 관심 변화를 배제하고, 임의의 시점에 전체 페이지에서 특정 페이지가 얼마나 요구되는지를 확률로 표현하며, zipf 분포[12]와 마찬가지로 특정한 일부 페이지에 요구가 집중된다. 반면에, 장기 인기도는 특정 페이지에 대한 사용자의 시간에 따른 관심의 변화 정도를 확률로 표현한다. [7]에서는 Pusanilbo's electronic의 기사에 대한 인기도 변화 패턴을 분석하였는데, 요약하면 그림 1, 그림 2와 같다.

그림 1은 단기 인기도 요구 패턴을 보여준다. 그림에서 y축은 인기도(즉, 접근 확률)를, x축은 접근 가능한 페이지들을 의미할 때, 각 페이지가 접근될 확률의 합 P는 식 (1)과 같이 표현할 수 있다.

$$P = \sum_{i=1}^{n} P(i) = 1 \qquad (1)$$

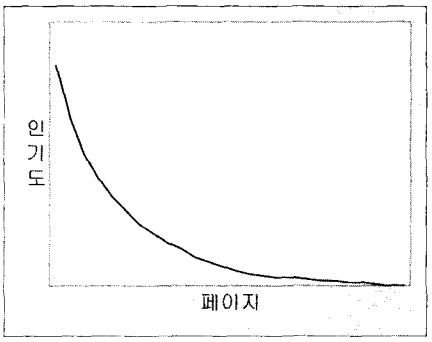

그림 1 단기 인기도 요구 패턴

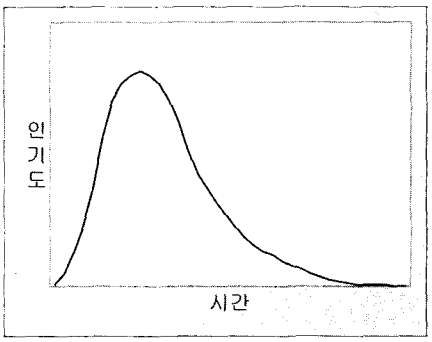

그림 2 장기 인기도 요구 패턴

여기서, n은 접근 가능한 페이지의 수를 의미하며, P(i)는 i번째 페이지가 접근될 확률이며 각 페이지의 접근 확률의 합은 1이 된다.

그림 2는 장기 인기도 패턴을 보여준다. y축은 인기도를, x축은 시간을 의미한다. 즉, 특정 페이지가 접근될 확률이 시간의 흐름에 따라 급속히 증가했다가 서서히 감소하는 형태를 보여준다. 이는 새로운 페이지가 데이타베이스에 삽입되었거나 혹은 클라이언트의 데이타 요구 변화가 있을 경우, 즉, 새로이 요구된 데이타의 경우 해당 페이지의 인기도가 그림 2와 같은 형태로 변화함을 나타내고 있다.

본 논문에서 인기도는 사용자의 요구 패턴의 변화에 따라 인기도가 그림 1과 그림 2의 패턴에 따르도록 유지한다. 페이지가 처음으로 요청되면, 해당 페이지의 우선순위(priority)는 임의의 큰 수로 초기화된다. 페이지가 요청될 때마다 우선순위를 1씩 증가시켜 자주 요청되는 페이지와 그렇지 못한 페이지간의 우선순위 차이를 반영한다. 반대로, 시간의 흐름에 따른 클라이언트의 데이타에 대한 관심의 변화를 반영하기 위해 우선순위를 감소시킨다. 우선순위가 감소하는 것은 해당 페이지에 대한 요청의 빈도가 이전에 비해 줄었음을 나타내므로 시간 t 간격마다 각 페이지의 요청 여부를 검사하여 그동안 요청되지 않은 페이지의 우선순위를 α(%) 만큼 감소시킨다. 이렇게하여 정해진 시간(t)안에 요청되지 않은 페이지는 향후에 점점 인기도가 낮아지고 요청된 페이지의 인기도는 상대적으로 높아짐을 반영할 수 있다. 인기도는 우선순위를 이용하여 다음 식 (2)와 같이 구한다. 구해진 인기도는 각 클라이언트의 페이지 접근 패턴을 보다 정확하게 반영하여 계산된 결과이므로 페이지 히트율을 높이는데 이용될 수 있다.

$$Popularity_i = \frac{P_i}{CT - RT_i} \qquad (2)$$

Popularityi는 페이지 i 의 인기도를 의미하며, P_i는 페이지 i의 우선 순위를, CT는 현재 시간, RT_i는 페이지 i의 요구 시간을 각각 의미한다. 분모는 동일한 우선순위를 갖는 페이지라도 최근에 요청된 페이지의 인기도를 높이기 위한 것이다. 따라서, 최근에 요청된 페이지일수록 분모값($CT-RT_i$)이 작게되어 상대적으로 높은 인기도를 갖게된다.

한편, α와 t의 결정 문제는 본 논문의 범위를 넘는 또 다른 최적화 문제이므로 본 논문에서는 임의로 α = 10%로 고정하였고 검사간격(t)은 임의로 정하기보다는 α = 10% 인 경우에 검사 간격(t)을 변화시키면서 클라이언트의 페이지 히트율을 구해 최대 히트율을 보인 t

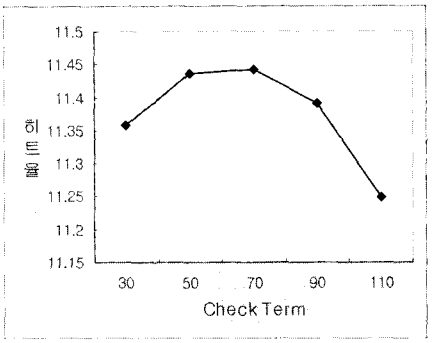

그림 3 검사 간격(t)에 의한 히트율의 변화 그래프

값을 선택하였다. 그림 3의 실험 결과에 따라 t-70(시뮬레이션 시간단위)으로 설정하였다.

2.2 대기 시간

대기 시간은 페이지에 대한 요구가 있은 후 실제 서버로부터 요구한 페이지를 받을 때까지 걸린 시간을 의미한다. 일반적으로 푸쉬 환경에서 서버는 접근 빈도나 인기도가 높은 페이지를 그렇지 않은 페이지보다 자주 방송하고, 풀 환경에서도 인기도가 높은 페이지는 자주 요청되기 때문에 자주 방송하게된다. 그림 4와 그림 5는 인기도 혹은 요청수가 높으면 이에 비례하여 대기시간은 감소한다는 가설을 그래프 형태로 보여주기 위한 것이다.

본 논문에서는 그림 4와 그림 5의 대기 시간을 반영하기 위해 식 (3)과 같이 가중 평균을 사용한다.

$$A(P_iW_{x+1}) = \frac{A(P_iW_x)\times\beta}{100} + \frac{P_iW_{x+1}\times(100-\beta)}{100} \quad (3)$$

식 (3)에서 P_iW_1는 특정 페이지 i 에 대한 최초 요구시 대기 시간을 의미하고, P_iW_x는 x번째 요구시 대기 시간을 의미한다. $A(P_iW_{x-1})$은 x+1 요구까지의 평균 대기 시간을 의미하며, P_iW_x까지의 평균 대기 시간을 β(%) 반영하고, P_iW_{x-1}의 대기 시간을 100-β(%) 반영하여 구한다. 이렇게 함으로써, 두 가지 인기도의 변화에 따른 대기 시간의 변화를 반영할 수 있으며, 비율 β를 조정하여 인기도의 변화에 따른 대기 시간의 변화를 좀 더 민감하게 반영할 수 있게 된다.

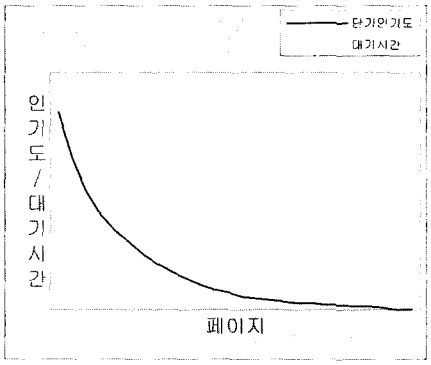

그림 4 단기 인기도에 따른 대기 시간의 변화 패턴

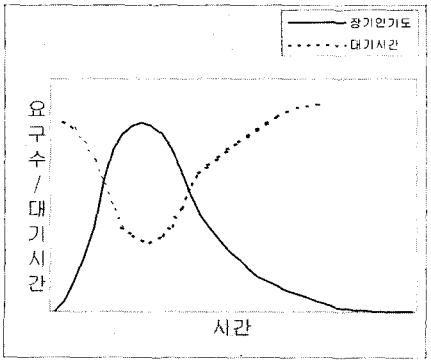

그림 5 장기 인기도에 따른 대기 시간의 변화 패턴

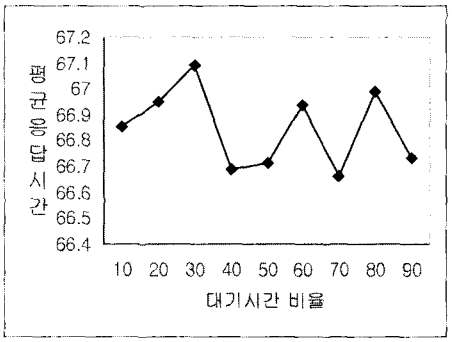

그림 6 반영 비율(β)에 의한 평균 응답 시간 변화 그래프

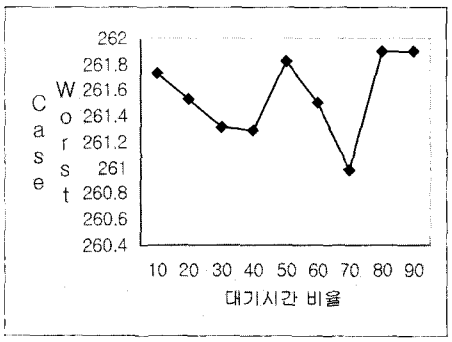

그림 7 반영 비율(β)에 의한 최악의 응답 시간 변화 그래프

본 논문에서 필요한 β 값을 임의로 설정하기보다는 나름대로 간단한 실험을 통해 설정하기 위하여 캐쉬 사이즈 100, 클라이언트 1000명이 요구하는 환경에서 β 값의 변화에 따른 평균 응답 시간의 변화를 실험하였다(그림 6과 그림 7). 그림에서 볼 수 있듯이 β의 값에 따라 응답 시간에 변화가 있으며, β 값이 70(%)에서 최소값을 보이고 있다. 따라서, 본 논문에서는 β 값을 70(%)로 고정하여 사용한다.

3. 캐쉬 대체 전략

3.1 동기

그림 1과 그림 2에서 보인 요구 패턴 변화와 관련하여 페이지 요구 변화와 대기 시간과의 상관관계를 통하여 캐쉬 대체 전략에서 미스비용을 고려할 필요성을 알아보기 위해 먼저 2000명의 클라이언트가 요구하는 풀 기반 환경에서 RxW 스케줄링 알고리즘[14]를 사용하여 실험하였다.

그림 8은 특정 시점에 요구된 페이지의 요구수를 나타내며 1번부터 300번 페이지까지는 빈번히 요구되고 나머지 페이지들은 그렇지 않은 경우를 보여주고 있다(그림 1 참조). 한편, 그림 9는 이러한 환경에서 각 페이지가 방송되기까지 큐(queue)에 대기한 대기 시간을 측정한 것으로 300번 페이지부터 응답 시간이 급격히 증가하는 모습을 볼 수 있다. 그림 8과 그림 9의 결과는 그림 4의 단기 인기도에 따른 대기 시간의 변화 패턴과 유사함을 확인할 수 있다.

한편, 그림 10은 특정 페이지가 시간의 흐름에 따라 요구된 요구수를 나타내며 시간의 흐름에 따라 급격히 요구수가 증가하여 일정 시간이 지나면 요구수가 차츰 감소하는 요구 변화를 보여준다(그림 2 참조). 그림 11은 대기시간은 요구수에 반비례한다는 가설에 대한 실험 결과로 그림 5와 모양에서는 다소 차이는 있으나 요구수가 감소함에 따라 대기시간은 증가함을 보여주고 있다.

이상의 실험으로부터, 이동 컴퓨팅 환경과 같이 대역의 제약이 따르는 환경에서는 대기 시간이 인기도와 상관관계가 있으므로 페이지 미스에 따른 대기 시간을 반영하기 위하여 미스 비용에 대한 고려가 필요함을 알 수 있다. 미스 비용을 고려하기 위해서, 페이지의 대기 시간만을 고려할 경우 미스 비용을 줄일 수 있지만, 인기도가 높은 페이지의 잦은 미스가 발생하여 성능 저하가 발생할 수 있다. 반대로 인기도만을 고려할 경우 히트율은 높일 수 있으나, 인기도는 상대적으로 작으나 대기 시간은 상대적으로 큰 페이지의 미스로 인해 성능

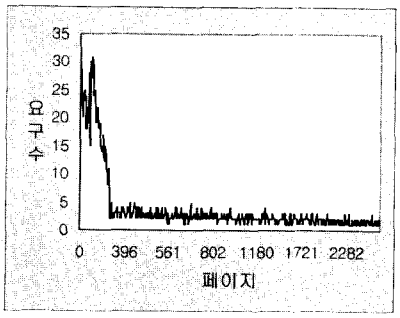

그림 8 실험을 통한 단기 인기도 요구 패턴의 변화

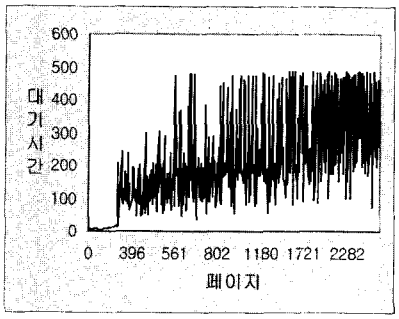

그림 9 실험을 통한 단기 인기도와 대기 시간의 상관관계

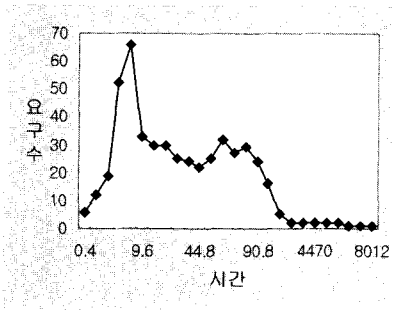

그림 10 실험에서의 장기 인기도 요구 패턴의 변화

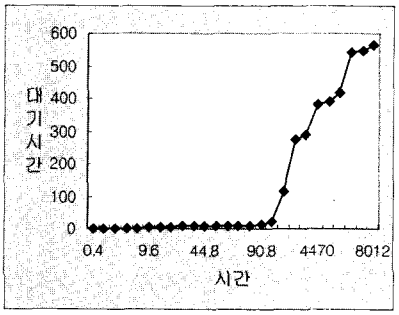

그림 11 실험에서의 장기 인기도와 대기 시간의 상관관계

저하가 발생할 수 있다.

3.2 캐쉬 대체 전략

이 절에서는 히트율과 미스 비용 모두를 고려하 대체 전략인 PWT(Popularity Waiting Time)를 제안한다. 히트율을 높이기 위해서는 인기도를 고려하여야 되고 미스 비용을 줄이기 위해서는 대기 시간을 고려하여야 하므로 PWT는 대체할 페이지를 선택하기 위해 인기도와 대기 시간의 곱한 값을 이용한다. 즉, PW값(PW = 인기도 × 대기 시간)을 구한 후에 가장 작은 PWT 값을 가지는 페이지를 대체할 페이지(victim)로 선택한다.

이를 위해 클라이언트는 4가지 속성으로 구성되는 PWT 테이블을 유지한다. 첫 번째는 페이지ID(PID)로 요청한 페이지의 식별자 역할을 한다. 두 번째는 페이지의 우선 순위 P로 2장에서 기술한 것처럼 우선 순위가 유지된다. 세 번째는 페이지가 마지막으로 요구된 시간이다. 네 번째는 대기 시간으로 2장에서 기술한 것처럼 유지된다.

인기도를 구하기 위해서는 우선순위(priority), 마지막으로 요구한 시간, 현재시간이 필요하므로(식 (2) 참조) 현재시간을 제외한 정보가 PWT 라는 테이블에 대기시간 정보와 함께 유지된다. 페이지가 처음으로 요청되면 해당 페이지는 임의의 수로 우선순위 초기값을 갖으면서 PWT 테이블에 등록된다. 캐쉬 내에 있는 페이지들에 대한 PW 값중에서 가장 낮은 값을 갖는 페이지가 희생자(victim)로 선택되므로 이들 값들은 별도의 최소 히프(min heap) 구조로 유지된다. 한편, 페이지 수가 많아져 PWT 테이블의 크기가 매우 커지게 되면 검색의 효율성을 위해 우선순위가 일정값 이하인 페이지들을 삭제하거나 해쉬 등의 방법을 고려할 수 있다.

한편, PWT 전략에서 PW 값을 구성하는 인기도 혹은 대기 시간을 1로 하면 각각 대기 시간 혹은 인기도만을 기반으로 하는 전략으로 귀착된다.

- 인기도 기반 대체 전략(P) : 페이지의 우선 순위와 요구 시간을 기반으로 식 (2)를 이용하여 인기도를 구한 후에, 가장 낮은 인기도를 갖는 페이지를 대체할 페이지로 선택하는 전략이다.
- 대기 시간 기반 대체 전략(WT) : 대기 시간을 유지하면서, 현 시점에 가장 작은 대기 시간을 갖는 페이지를 대체할 페이지로 선택하는 전략이다.

그림 12는 캐쉬 대체 전략의 예를 위한 표로 현재 시점까지 요구를 수용한 결과 실제 캐쉬에 있는 페이지의 인기도와 대기 시간을 구한 것이다. 현재 캐쉬에서 대체 페이지 선정을 위해서 대체 전략 P를 적용하면 가장 작은 인기도를 갖는 페이지 f(인기도-5)가 대체할 페이지로 선정되며 대체 전략 WT를 적용하면 가장 작은 대기 시간을 갖는 페이지 a(대기 시간-15)가 대체할 페이지로 선정된다. 한편, 대체 전략 PWT를 적용하면 가장 작은 PW값을 갖는 페이지 b(PW-140)가 대체할 페이지로 선정된다.

PID	인기도	대기 시간
a	25	15
d	21	18
f	5	30
c	15	19
e	10	20
b	7	20

그림 12 캐쉬 대체 전략 예제

4. 성능 평가

4.1 성능평가 모델

본 논문에서는 기존 연구에서와 같이 기본적인 몇가지 공통된 가정을 한다.

- 데이타 크기는 일정하다. 즉, 데이타베이스의 데이타는 동일 크기의 데이타 항목 단위로 나뉘어 지며, 각 항목 단위로 서비스가 이루어진다고 가정한다.
- 데이타는 읽기 전용이고 일관성 고려는 하지 않는다. 클라이언트가 자체 캐쉬를 가짐으로써, 클라이언트와 서버 사이의 데이타 일관성 문제가 발생할 수 있다. 본 논문에서는 데이타가 읽기 전용으로 클라이언트와 서버 사이의 일관성 문제는 또 다른 연구 방향이므로 고려하지 않는다.
- 클라이언트는 요구한 데이타를 수신 받을 때까지 새로운 데이타를 요구하지 않는다. 일정 시점에 데이타 요구는 단일 항목에 국한되며, 먼저 요구한 데이타를 수신 받기 전에 새로운 요구를 발생시키지 않는다. 이는 복잡한 시뮬레이션 환경을 단순화하기 위한 가정이다.
- 방송은 단일 채널을 통해서 이루어진다. 클라이언트는 지속적으로 하나의 채널을 모니터링하며 배터리 소모는 고려하지 않는다.
- 무선 환경에서 빈번히 발생할 수 있는 전송 에러는 고려하지 않는다.

본 논문에서 기반으로 하는 성능 평가 모델은(그림 13), 크게 데이타를 요구하는 이동개체(Mobile Unit :MU)인 클라이언트와 요구를 수용하여 방송하는 고정 호스트(fixed host) 및 이동기지국(Mobile Service Provider:MSS)인 서버로 나눌 수 있으며, 운영 환경 모

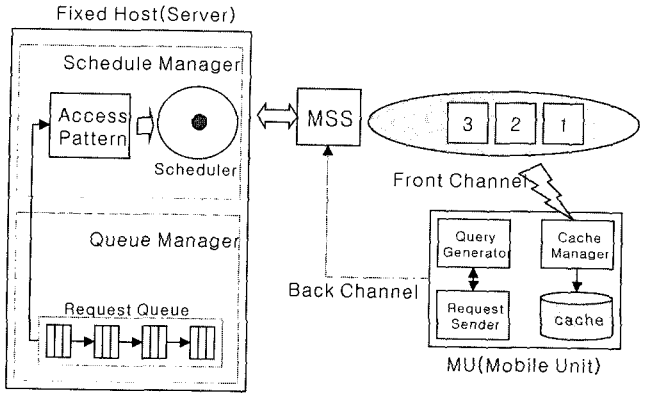

그림 13 성능 평가 모델

표 2 운영 환경 모델 파라미터

파라미터	내 용	단위
SIMTIME	시뮬레이션시간	Time(시뮬레이션 시간)
Frontchannel	데이타 방송채널 대역폭 크기	Bps
Backchannel	데이타 요구채널 대역폭 크기	Bps
Data_transfer_time	데이타 전송시 걸리는 시간	Time(시뮬레이션 시간)
No_Backchannel	동시 전송 가능한 Backchannel 수	클라이언트 수
Upload_Time	서버쪽으로 데이타 전송시간	Time(시뮬레이션 시간)

델 파라미터는 표 2와 같다.

본 논문에서 풀 기반의 데이타 방송은 Aksoy가 제안한 RxW 스케쥴링 알고리즘을 사용하여 요구수와 큐에 대기한 대기 시간을 기반으로 한다. 이를 위해 서버는 큐 관리자(queue manager)와 스케쥴 관리자(schedule manager)의 두 가지 구성 요소를 가지며, 서버 파라미터는 표 3과 같다.

4.1.1 큐 관리자(queue manager)

서버의 큐는 클라이언트에 의해 요청된 데이타를 분류하고 저장하는 역할을 수행한다. 데이타의 요청을 받은 서버 큐는 서버 큐 내에 요구된 데이타가 이미 존재하는지를 검사하게 된다. 이 과정에서 중복된 데이타는 요구 빈도값을 증가시킨다. 존재하지 않는 데이타의 경우에는 새로운 노드를 생성하여 서버 큐에 삽입하게 된다.

4.1.2 스케쥴 관리자(schedule manager)

큐 관리자에서 관리하고 있는 데이타의 방송을 위한 데이타 선정 작업과 제시된 스케쥴링 방식(RxW)에 따라서 데이타 방송 순서를 결정하는 역할을 한다. 스케쥴링 되어진 데이타는 방송 대역폭인 다운링크를 통해 순차적으로 전송되며, 데이타는 방송 수신 상태에 있는 모든 클라이언트에게 전송된다.

표 3 서버 파라미터 기술

변수명	내 용	단 위
DBSize	서비스 데이타베이스 크기	Item
AccessRange	서비스 Item 크기	Item

한편, 풀 환경에서는 클라이언트가 방송할 데이타를 요구하므로 클라이언트는 질의 생성기를 포함하여 다음과 같이 세 부분으로 구성되며, 클라이언트 파라미터는 표 4와 같다.

4.1.3 질의 생성기(query generator)

클라이언트는 성능평가 모델에서 개별 프로세스 단위로 동작하게 된다. 클라이언트 프로세스의 첫 동작은 클라이언트 데이타 요구 단계로, 질의 생성기로부터 시작된다. 질의 생성기는 그림 8과 그림 9에서 보인 단기 인기도와 장기 인기도를 반영하여 데이타에 대한 요구를 한다. 본 논문에서는 클라이언트의 요구 변화 패턴을 반영하기 위해 Frequency와 Offset 변수를 사용하였다. Frequency는 클라이언트의 요구 변화 빈도를 Offset은 요구 변화 정도를 나타낸다. 클라이언트 프로세스는 Frequency 값에 따라 데이타를 변경하여 요청할 때

표 4 클라이언트 파라미터 기술

변수명	내 용	단 위
Number_Client	서비스 사용자수	Number/Cell
CacheSize	사용자 캐쉬 크기	Item
Cache_Access_time	캐쉬 액세스 시간	Time(시뮬레이션 시간)
ThinkTime	방송을 기다리는 시간	Time(시뮬레이션 시간)
Check_Term	Popularity 값 조정 간격	Time(시뮬레이션 시간)
Pop_Ratio	Popularity 조정 값	%
Wait_Ratio	Wait Time 반영 비율	%
PWTSIZE	PW Table의 사이즈	Item
Frequency	데이타 요구변경 주기	Time(시뮬레이션 시간)
Offset	데이타 요구변경 범위	Item
INTARV	다음 질의까지 시간 간격	Time(시뮬레이션 시간)

Offset 만큼 떨어진 데이타를 요구하게 된다.

4.1.4 요청 송신기(request sender)

만일 캐쉬에 요구 데이타가 없을 경우 클라이언트 프로세스는 일정기간(ThinkTime)동안 서버측으로부터 방송되는 내용을 청취한다. 이 과정은 서버측으로 요구를 전달할 수 있는 대역폭이 상대적으로 적기 때문에 대역폭의 효율적 이용을 위한 측면과, 서버측에 과도한 요구와 경쟁을 감소시키는 측면에서 효과를 거둘 수 있다. 이 과정에서도 얻지 못한 데이타는 서버에 직접 요청하는 과정을 거쳐 데이타가 방송되면 데이타를 수신하게 되고, 다음 질의까지 기간(INTARV) 동안 새로운 데이타를 요구하지 않게 된다.

4.1.5 캐쉬 관리자(cache manager)

클라이언트는 질의한 데이타의 수신을 위해 방송되는 데이타 중에서 해당 데이타가 방송되는지를 지속적으로 모니터링 한다. 이 과정에서 서버측으로부터 데이타를 방송받는데 걸리는 시간을 반영하여 검사한다. 캐쉬 관리자는 요청한 데이타를 수신하면 캐쉬에 데이타의 저장을 시도한다. 이 과정에서 캐쉬 대체 전략이 적용된다.

4.2 변수 설정

일반적으로 방송 채널의 크기는 기존 연구에서 10kbps~19.2kbps로 설정하고 있다. 본 논문에서는 19.2 kbps 대역폭을 가지는 방송채널을 고려하였으며, 대역폭의 비대칭성에 관한 고려를 적용한 기존 연구의 제시에 따라 방송 대역폭의 1%로 요구 대역폭의 크기를 설정하였다. 동시에 수용 가능한 요구 데이타는 30개로 클라이언트 30명이 동시에 요구 데이타를 전송하게 되면, 31번째 클라이언트는 전송이 끝나기를 기다려야 하며, 점유된 요구채널은 고정된 전송 시간(Data_transfer_time)이 소요된 후에 다음 요구 데이타의 전송을 수용할 수 있다. 또한 시뮬레이션 시간은 총 20000 시간 동안 실험하였지만, 이중 10000 시간은 캐쉬와 PWT 테이블을 채우기 위한 예비 시간으로 시뮬레이션 결과에 포함시키지 않았다.

표 5 운영 환경 모델 파라미터 설정

파라미터	설정값	단위
SIMTIME	10000	Time(시뮬레이션 시간)
Frontchannel	19200	Bps
Backchannel	192	Bps
Data_transfer_time	0.2	Time(시뮬레이션 시간)
No_Backchannel	30	클라이언트 수
Upload_Time	0.01	Time(시뮬레이션 시간)

표 6 서버 파라미터 설정

변수명	설정값	단 위
DBSize	5000	Item
AccessRange	3000	Item

데이타베이스 크기(DBSize)는 기존 논문에서 보편적으로 사용한 3000개의 서비스 항목을 중심으로 1000, 3000, 5000개의 서비스 데이타 항목수를 가지는 데이타베이스 환경을 설정하였다. 본 연구에서는 5000개의 데이타베이스 크기를 가지며, 특정 시점에 클라이언트가 데이타베이스에 관심을 가지는 범위를 3000개로 하였다.

각 클라이언트는 시뮬레이션 프로그램의 시작과 동시에 클라이언트 수(Number_Client)만큼 개별 프로세스로 설정되어 독립적인 클라이언트의 역할을 수행한다. 방송 수신 가능 영역인 셀 내의 클라이언트의 수는 방송 항목의 수에 따른 비율로 1000명까지로 설정하였다. 각 클라이언트는 캐쉬를 소유하고 있으며, 캐쉬의 크기

표 7 클라이언트 파라미터 설정

변수명	설정값	단 위
Number_Client	10, 50, 100, 200, 300, 500, 700, 1000	Time(시뮬레이션 시간)
CacheSize	10, 50, 100, 150, 200, 300	Item
Cache_Access_time	0.001	(1Item/Time)
ThinkTime	2	Time(시뮬레이션시간)
Check_Term	30, 50, 70, 90, 110	Time(시뮬레이션시간)
Pop_Ratio	10, 30, 50, 70, 90	%
Wait_Ratio	10, 30, 50, 70, 90	%
PWTSIZE	캐쉬 사이즈와 동일	Item
Frequency	0 1000	Time(시뮬레이션시간)
Offset	0 80	Item
INTARV	Random(0, 3)	Time(시뮬레이션 시간)

〈CacheSize〉는 10-300 항목을 저장할 수 있고 크기를 변화시켜 가며 실험을 진행하였다.

클라이언트의 데이타 요구 패턴을 계산하기 위하여 인기도 조정 간격을 30-110으로 설정하고, 인기도와 대기 시간의 반영비율은 10 - 90(%)로 설정하여 실험하였다. 또한, 요구 변화 빈도와 변화 정도는 LRU, P, WT, PWT 간의 외부 요인을 배제하기 위해서 요구 변화 빈도는 1000(time)으로 변화 정도는 80(item)으로 고정하였다.

4.3 결과 분석

제안한 대체전략 PWT와 이의 변형인 P, WT, 그리고 LRU에 대한 성능 비교를 히트율, 평균 응답시간, 그리고 최악 응답시간을 척도로 성능을 비교하였다. 성능 평가를 위해 요구 패턴 변화 빈도와 변화 정도를 각각 1000 과 80으로 고정시켰다. 요구 패턴 변화 빈도와 정도를 랜덤 함수로 산출하는 경우, 성능이 이에 종속될 수 있으므로 종속성을 최소화하기 위한 것이다. 변화 빈도를 고정시키는 대신에 클라이언트 수의 변화에 따라 요구 패턴 주기가 변화하는 효과를 얻도록 하였다. 바꾸어 말하면, 클라이언트 수가 증가하면 각 페이지에 대한 응답 시간이 커져서 각 클라이언트가 요구 패턴 변화 빈도주기(1000 시간) 동안 요구할 수 있는 요구 횟수가 적어지게 되어, 각 클라이언트 입장에서 보면 소수의 요청후에 요구 패턴 변화가 발생하므로 상대적으로 요구 변화가 심한 효과 즉, 변화 시간 간격이 줄어드는 효과를 얻게 된다. 따라서, 클라이언트 수가 적은 경우는 요구 변화 주기 시간이 상대적으로 큰 효과를 볼 수 있으며, 클라이언트 수가 큰 경우는 요구 변화 주기 시간이 상대적으로 작은 효과를 볼 수 있다.

4.3.1 히트율

그림 14는 클라이언트의 수가 상대적으로 적은 환경으로, 요청에 대한 응답 시간이 상대적으로 적기 때문에, 요구 패턴의 변화가 일어나기 전에 보다 많은 요청을 하는 결과를 초래한다. 따라서, 이러한 환경에서는 인기도가 높은 페이지가 자주 요청되어 히트율이 상대적으로 높게 나타난다. 따라서, 캐쉬 크기와 상관없이 가장 좋은 성능을 보인 대체 전략은 인기도를 고려한 P와 LRU 대체 전략이 된다.

그림 15와 그림 16은 그림 14보다 많은 수의 클라이언트를 가진 환경으로, 요청에 대한 응답 시간이 상대적으로 커지기 때문에, 클라이언트의 요구 패턴 변화가 발생하기까지 클라이언트의 요구수가 상대적으로 작아지므로 요구 변화 주기가 짧아지는 환경으로 볼 수 있다. 이러한 환경에서 가장 좋은 성능을 보인 대체 전략은 그림 14와 마찬가지로 LRU와 P이지만, 캐쉬 크기가 커지면서 격차가 줄어드는 모습을 보인다. 이는 클

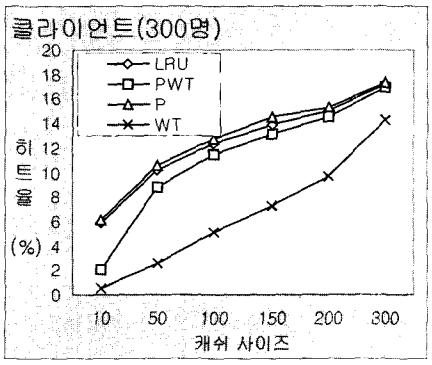

그림 14 히트율(클라이언트 : 300)

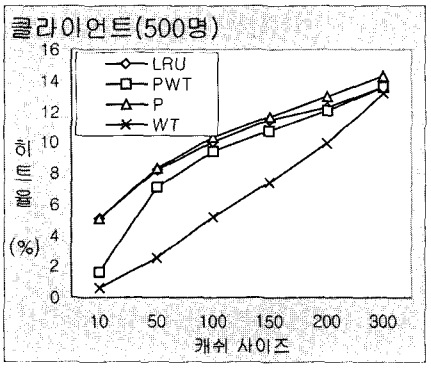

그림 15 히트율(클라이언트 : 500)

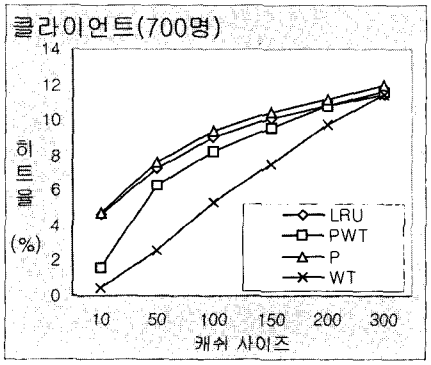

그림 16 히트율(클라이언트 : 700)

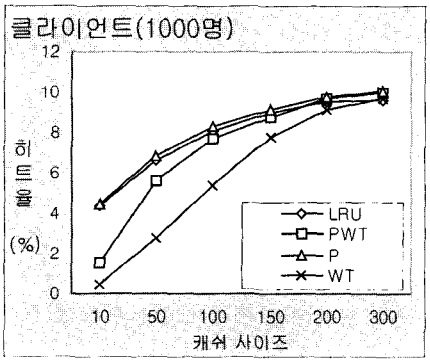

그림 17 히트율(클라이언트 : 1000)

라이언트의 요구 변화 주기가 보다 짧아지면서, 현시점에 인기도가 높은 페이지가 캐쉬에 있어서 발생하는 캐쉬 히트보다는 인기도가 상대적으로 낮은 페이지의 캐쉬 히트도 상대적으로 커지는 결과를 반영한 것으로 보인다.

그림 17은 클라이언트가 가장 많은 경우로, 요구 패턴 변화가 상대적으로 가장 빈번한 경우를 반영한 결과라고 할 수 있다. 그림 15와 그림 16과 같이 캐쉬 크기가 증가하면서 대체 전략간의 격차가 줄어듬을 알 수 있다. 뿐만 아니라, 가장 빈번한 요구 변화로 인하여 가장 낮은 히트율을 보이고 있음을 알 수 있다.

요약하면, 히트율 측면에서 인기도를 기반으로 하는 P와 LRU가 PWT 보다 우월한 성능을 보이고 있으며 대기 시간만을 고려하는 WT는 페이지의 인기도가 가장 작은 페이지가 캐쉬에 있을 가능성이 크므로 가장 낮은 성능을 보인다.

결국, 캐쉬 크기와 요구 변화 패턴의 주기 간격에 따라 효율적인 대체전략을 생각해 볼 수 있다. 캐쉬 크기가 상대적으로 작은 경우는 히트율을 높이기 위해서 LRU와 대체 전략 P를 사용하는 것이 가장 좋은 성능을 보이며 이는 캐쉬 크기가 클 경우도 마찬가지다. 그러나, 캐쉬 크기가 크며 요구 패턴 변화 주기가 상대적으로 짧은 환경에서는 히트율의 크기가 거의 비슷한 결과를 보였으므로, 평균 응답 시간과 같은 다른 기준을 고려하는 것이 의미가 있음을 말해 주고 있다.

4.3.2 평균 응답 시간

클라이언트의 수가 적을수록, 서버측에서 요구하는 페이지에 대한 경쟁이 줄어들기 때문에, 요청에 대한 응답 시간은 상대적으로 향상되는 결과를 초래한다(그림 18~그림 21). 또한, 대기 시간은 페이지가 서버의 큐에 대기한 시간을 의미하는 것이므로 대기 시간이 큰 페이지는 작은 페이지보다 응답 시간이 클 수밖에 없다. 이렇게 대기 시간이 큰 페이지가 캐쉬에 있을 경우에 히트를 하게 되면, 대기 시간이 작은 페이지보다 응답 시간이 크게 줄어드는 결과를 보이게 되어 마찬가지로 평균 응답 시간은 향상된다. 한편, 클라이언트 수가 많은

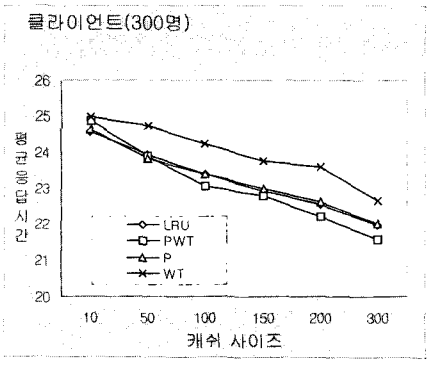

그림 18 평균응답시간(클라이언트 300)

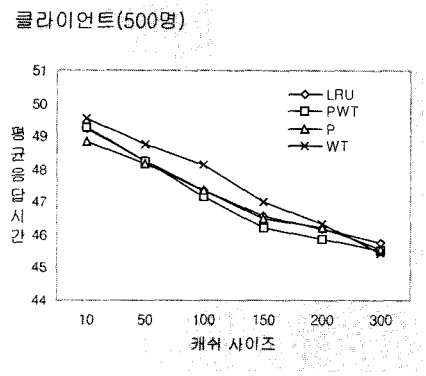

그림 19 평균응답시간 (클라이언트 500)

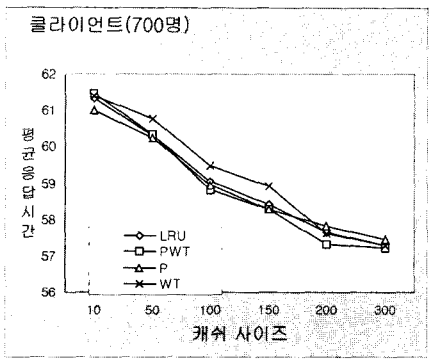

그림 20 평균응답시간(클라이언트 700)

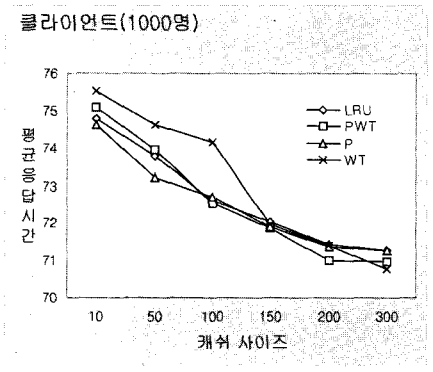

그림 21 평균응답시간(클라이언트 1000)

경우에 캐쉬 크기가 절대적으로 부족하면 히트율의 차이가 커져 대기시간을 고려한 효과를 보기 힘들게 되어 평균 응답시간에서 우위를 보이지 못하지만 캐쉬 크기가 커질수록 평균 응답시간은 향상되어 대기시간을 고려한 효과를 확인할 수 있다. 따라서, 대기 시간과 인기

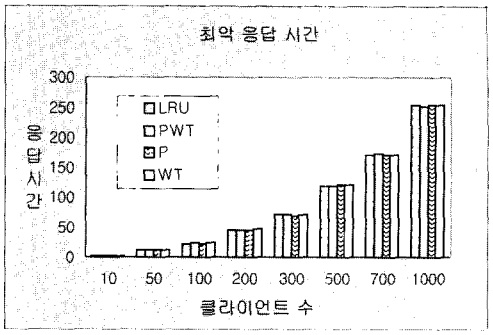

그림 22 최악응답시간(클라이언트 1000)

도를 같이 고려한 PWT가 다른 전략에 비해 성능면에서 우위에 있게됨을 알 수 있다.

한편, 캐쉬 크기가 작을 경우에는 캐쉬 대체 전략간의 히트율이 차이가 컸다(그림 14 참고). 즉, 히트율의 차이가 크므로, 히트율이 가장 좋은 대체 전략이 보다 높은 성능을 보이게 된다. 그러나, 캐쉬 크기가 커지면서 히트율의 차이가 줄어들게 됨에 따라 요구 패턴 변화 주기가 짧은 경우에는 대기 시간을 고려하는 것이 응답시간을 줄이는 측면에서 유리해진다. 이는 그림 21로부터 확인할 수 있다. 그림에서 보면, 요구 패턴 변화 주기가 상대적으로 짧아지면서 WT가 다른 전략들과의 성능 차이를 줄여 가고 있음을 볼 수 있다. 이는 대체 전략의 히트율의 차이가 거의 없을 경우 대기 시간이 큰 페이지를 캐쉬에 유지하는 WT가 응답 시간에서 유리해짐을 반영한 결과로 볼 수 있다.

요약하면, 히트율의 차이가 비슷한 환경에서는 성능평가 척도로 평균응답시간을 고려할 필요가 있다. 이 경우 인기도와 대기시간을 함께 고려하는 전략이 전반적으로 우위에 있음을 알 수 있다.

4.3.3 최악 응답 시간

최악의 응답 시간은 각 클라이언트의 요구에 대해서 가장 최악의 응답 시간의 평균을 구한 것으로, 모든 캐쉬 대체 전략에서 비슷한 결과를 얻었다(그림 22). 이는 방송 스케쥴링 기법인 RxW의 특징으로 요청이 극히 적은 페이지에 대해서도 최악의 응답 시간을 보장할 수 있기 때문으로 생각된다. 앞에서 설명한 그림 3-2의 응답 시간의 변화 추이를 보면, 2000명의 클라이언트를 가지고 있는 상황에서도 최악의 응답 시간은 500을 넘지 않는 것을 확인할 수 있다.

5. 결 론

본 논문에서는 이동 컴퓨팅 환경에서 요구 변화가 비

교적 심한 풀 기반 환경에서 적용할 수 있는 대체 전략을 제안하였다. 제안한 캐쉬 대체 전략 PWT는 클라이언트 측면에서 사용할 수 있는 요구 패턴 분석을 기반으로 한다. 이를 위해 인기도와 페이지 대기 시간을 고려하였으며 PWT는 인기도와 대기 시간을 함께 고려하여 히트율을 높이고 미스 비용을 줄이려고 하였다.

제안한 캐쉬 대체 전략의 성능 검증을 위해 LRU, P, WT, PWT의 4가지 대체 전략을 시뮬레이션을 통해 히트율, 평균 응답 시간, 최악 응답 시간의 세 가지 척도를 사용하였다. 히트율에서는 LRU와 P가 요구 패턴의 변화가 적고 캐쉬가 적은 환경에서 좋은 성능을 보였으나, 요구 변화가 상대적으로 심하고 캐쉬가 상대적으로 큰 환경에서는 성능 차이가 줄어들었다. 평균 응답 시간에서는 PWT가 전체적으로 좋은 성능을 보였으며, 요구 변화가 심한 환경이 될수록 대기 시간의 고려가 필요해짐에 따라 WT는 PWT와 비슷한 성능을 보였다. 최악 응답 시간은 전체적으로 비슷한 결과를 보였는데, 이는 서버 방송 스케쥴링인 RxW의 특징에 기인한 것이다.

참 고 문 헌

[1] D. Aksoy, M. Altinel, R. Bose, U. Cetintemel, M. Franklin, J. Wang, S. Zdonik, "Research in Data Broadcast and Dissemination," AMCP, pp. 194 207, 1998.

[2] D. Barbara, "Mobile Computing and Databases A Survey," IEEE Transactions on Knowledge En gineering, Vol. 11, No. 1, pp. 108 117, January /February 1999.

[3] M. Frnaklin, S. Zdonik, "Dissemination Based Information Systems," IEEE Data Engineering Bulletin, Vol. 19, No.3, pp. 20 30, Sept. 1996.

[4] S. Acharya, M. Franklin, S. Zdonik, "Balancing Push and Pull for Data Broadcast," Proc. of ACM SIGMOD, Tuscon, Arizona, pp. 183 194, May 1997.

[5] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, "Broadcast Disks : Data Management for Asymmetric Communication Environments," Proc. of ACM SIGMOD, pp. 199 210, 1995.

[6] C. Griwodz, M. Bär, L. C. Wolf, "Long term Movie Popularity Models in Video on Demand Systems or The Life of an on Demand Movie," ACM Int. Conf. on Multimedia, Seattle, USA, pp. 349 357, 1997.

[7] T. Choi, Y. Kim, K. Chung, "A prefetching scheme based on the analysis of user access patterns in news-on demand system," Proc. of the 7th ACM Int. Conf. on Multimedia, pp. 145 - 148, 1999.

[8] S. Khanna, V. Liberatore, "On Broadcast Disk Paging," Proc. of the 30th ACM Symp. on the Theory of Computing, pp. 634 643, 1998.

[9] V. Liberatore, "Caching and Scheduling for Broadcast Diks Systems," Technical Report 98-71, UMIACS, 1998.

[10] S. Galvin, P. B. Galvin, Operation System Concepts, 4th Edition, Addison Wesley, 1994.

[11] E. O'Neil, P. O'Neil, G. Weikum, "The LRU K page replacement algorithm for database disk buffering," Proc. of ACM SIGMOD, pp. 297 306, May 1993.

[12] D. Knuth, The Art of Computer Programming, Vol II, Addison Wesley, 1981.

[13] D. Lee, J. Choi, J. Kim, S.Noh, S. Min, Y. Cho, C. Kim, "On the Existence of a Spectrum of Policies that Subsumes the Least Recently Used(LRU) and Least Frequently Used(LFU) Policies," Proceedings of ACM SIGMETRICS'99(International Conference on Measurement and Modeling of Computer Systems), pp. 134 143, 1999.

[14] D. Aksoy, M. Franklin, "RxW : A Scheduling Approach for Large Scale On-Demand Data Broadcast," IEEE/ACM Transactions on Networking Vol. 7, No. 6, pp. 846 860, 1999.

이 윤 장

1998년 중앙대학교 산업정보학과 학사
2000년 중앙대학교 시스템통합과정 수료
2002년 중앙대학교 정보시스템학과 석사
관심분야는 이동 컴퓨팅, 데이타베이스 시스템

신 동 천

1985년 2월 서울대학교 공과대학 컴퓨터공학과 졸업(학사). 1987년 2월 한국과학기술원 전산학과 졸업(석사). 1991년 2월 한국과학기술원 전산학과 졸업(박사) 1991년 1월~1993년 2월 한국전산원 선임연구원. 1993년 3월~현재 중앙대학교 산업과학대학 정보시스템학과 교수. 관심분야는 이동 데이타베이스, 다중 데이타베이스, 데이타 웨어하우스